CLAIM AMENDMENT

1. (Currently Amended) A method for intelligently caching applications and data on a gateway, comprising the steps of:

calculating a cache benefit index for a set of files, said cache benefit index:

being calculated based at least on the frequency of updates of said set of files at said gateway; and

indicating a total benefit for caching said set of files;

determining whether to cache said set of files on a local file system based on said cache benefit index;

caching said set of files on said local file system; and

updating a set of tables in a gateway database based on said caching.

2. (Original) The method of claim 1, wherein said determining includes the steps of:

checking available free space in said local file system; and

allowing caching of said set of files into said local file system if said local file system has enough available free space for storing said set of files.

3. (Original) The method of claim 1, wherein said determining includes the steps of:

comparing said cache benefit index to a previously calculated cache benefit index for said set of files; and

allowing caching of said set of files if said cache benefit index is higher than said previously calculated cache benefit index.

4. (Original) The method of claim 1, wherein said determining includes the steps of:

comparing said cache benefit index to other cache benefit indices of files already cached on said local file system; and

allowing caching of said set of files if said cache benefit index is higher than said other cache benefit indices.

5. (Original) The method of claim 1, further comprising the steps of:

recalculating a new cache benefit index for said set of files upon receiving a request to download or update said set of files; and

updating said set of tables in said gateway database based on said new cache benefit index.

6. (Original) The method of claim 1, further comprising the steps of:

receiving a download request for said set of files;

accessing said set of files in said local file system if said set of files is cached and up-to-date;

creating a download response to said download request, said download response including said set of files; and

sending said download response.

7. (Original) The method of claim 6, further comprising the step of:

downloading said set of files from a server if it is not cached.

8. (Original) The method of claim 7, wherein said downloading includes the steps of:

sending a request to said server;

receiving a response from said server, said response including said set of files;

parsing said response for a broadcast message;

accessing and updating said gateway database if said response includes said broadcast message; and

sending a broadcast response to said server.

9. (Original) The method of claim 6, further comprising the step of:

updating said set of files from a server if it is not up-to-date.

10. (Original) The method of claim 1, further comprising the steps of:

receiving an update request for said set of files;

accessing said local file system to obtain at least one difference file and a broadcast message if said set of files is cached and up-to-date;

creating an update response to said update request, said update response including said at least one difference file and said broadcast message; and

sending said update response.

11. (Original) The method of claim 1, further comprising the steps of:

receiving an update request for said set of files;

downloading said set of files from a server if said set of files is not cached;

creating an update response to said update request, said update response including said downloaded set of files; and

sending said update response.

12. (Original) The method of claim 1, further comprising the steps of:

receiving an update request for said set of files;

receiving at least one difference file from a server if said set of files is cached but is not up-to-date;

creating an update response to said update request, said update response including said at least one difference file; and

sending said update response.

13. (Original) The method of claim 1, further comprising the steps of:

receiving an update request for said set of files;

downloading a current version of said set of files from a server if said set of files is cached but is not up-to-date;

generating at least one difference file based on said current version;

creating an update response to said update request, said update response including said at least one difference file; and

sending said update response.

14. (Original) The method of claim 1, further comprising the steps of:

receiving a status check request for said set of files;

accessing said local file system to load any broadcast information if said set of files is up-to-date;

creating a status check response, said status check response including said broadcast information and a status of said set of files; and

sending said status check response.

15. (Original) The method of claim 1, further comprising the steps of:

receiving a status check request for said set of files;

sending a request to a server if said set of files is cached and is not up-to-date;

receiving a server response from said server, said server response including a current version and status of said set of files;

updating said gateway database based on said current version and status;

creating a status check response, said status check response including said status of said set of files; and

sending said status check response.

16. (Original) The method of claim 15, further comprising the steps of:

generating at least one difference file based on said current version;

updating said set of files based on said difference file; and

sending said difference file in said status check response.

17. (Original) The method of claim 15, further comprising the steps of:

parsing said server response for any broadcast message;

accessing and updating said gateway database if said server response includes a broadcast message; and

sending a broadcast response to said server.

18. (Original) The method of claim 1, further comprising the steps of:

receiving a status check request for said set of files;

downloading a current version of said set of files from a server;

comparing said current version of said set of files to said set of files;

generating said status based on said comparing;

creating a status check response, said status check response including said status of said set of files; and

sending said status check response.

19. (Currently Amended) A computer program product for use in conjunction with a computer system for intelligently caching applications and data on a gateway, comprising:

logic code for calculating a cache benefit index for a set of files, said cache benefit index:

being calculated based at least on the frequency of updates of said set of files at said gateway; and

indicating a total benefit for caching said set of files;

logic code for determining whether to cache said set of files on a local file system based on said cache benefit index;

logic code for caching said set of files on said local file system; and

logic code for updating a set of tables in a gateway database based on said caching.

20. (Original) The computer program product of claim 19, wherein said logic code for determining includes:

logic code for checking available free space in said local file system; and

logic code for allowing caching of said set of files into said local file system if said local file system has enough available free space for storing said set of files.

21. (Original) The computer program product of claim 19, wherein said logic code for determining includes:

logic code for comparing said cache benefit index to a previously calculated cache benefit index for said set of files; and logic code for allowing caching of said set of files if said cache benefit index is higher than said previously calculated cache benefit index.

22. (Original) The computer program product of claim 19, wherein said logic code for determining includes:

logic code for comparing said cache benefit index to other cache benefit indices of files already cached on said local file system; and

logic code for allowing caching of said set of files if said cache benefit index is higher than said other cache benefit indices.

23. (Original) The computer program product of claim 19, further comprising:

logic code for recalculating a new cache benefit index for said set of files upon receiving a request to download or update said set of files; and

logic code for updating said set of tables in said gateway database based on said new cache benefit index.

24. (Original) The computer program product of claim 19, further comprising:

logic code for receiving a download request for said set of files;

logic code for accessing said set of files in said local file system if said set of files is cached and up-to-date;

logic code for creating a download response to said download request, said download response including said set of files; and

logic code for sending said download response.

25. (Original) The computer program product of claim 24, further comprising:

logic code for downloading said set of files from a server if it is not cached.

26. (Original) The computer program product of claim 25, wherein said logic code for downloading includes:

logic code for sending a request to said server;

logic code for receiving a response from said server, said response including said set of files;

logic code for parsing said response for a broadcast message;

logic code for accessing and updating said gateway database if said response includes said broadcast message; and

logic code for sending a broadcast response to said server.

27. (Original) The computer program product of claim 24, further comprising:

logic code for updating said set of files from a server if it is not up-to-date.

28. (Original) The computer program product of claim 19, further comprising:

logic code for receiving an update request for said set of files;

logic code for accessing said local file system to obtain at least one difference file and a broadcast message if said set of files is cached and up-to-date;

logic code for creating an update response to said update request, said update response including said at least one difference file and said broadcast message; and

logic code for sending said update response.

29. (Original) The computer program product of claim 19, further comprising:

logic code for receiving an update request for said set of files;

logic code for downloading said set of files from a server if said set of files is not cached;

logic code for creating an update response to said update request, said update response including said downloaded set of files; and

logic code for sending said update response.

30. (Original) The computer program product of claim 19, further comprising:

logic code for receiving an update request for said set of files;

logic code for receiving at least one difference file from a server if said set of files is cached but is not up-to-date;

logic code for creating an update response to said update request, said update response including said at least one difference file; and

logic code for sending said update response.

31. (Original) The computer program product of claim 19, further comprising:

logic code for receiving an update request for said set of files;

logic code for downloading a current version of said set of files from a server if said set of files is cached but is not up-to-date;

logic code for generating at least one difference file based on said current version;

logic code for creating an update response to said update request, said update response including said at least one difference file; and

logic code for sending said update response.

32. (Original) The computer program product of claim 19, further comprising:

logic code for receiving a status check request for said set of files;

logic code for accessing said local file system to load any broadcast information if said set of files is up-to-date;

logic code for creating a status check response, said status check response including said broadcast information and a status of said set of files; and

logic code for sending said status check response.

33. (Original) The computer program product of claim 19, further comprising:

logic code for receiving a status check request for said set of files;

logic code for sending a request to a server if said set of files is cached and is not up-to-date;

logic code for receiving a server response from said server, said server response including a current version and status of said set of files;

logic code for updating said gateway database based on said current version and status;

logic code for creating a status check response, said status check response including said status of said set of files; and logic code for sending said status check response.

34. (Original) The computer program product of claim 33, further comprising:

logic code for generating at least one difference file based on said current version;

logic code for updating said set of files based on said difference file; and

logic code for sending said difference file in said status check response.

35. (Original) The computer program product of claim 33, further comprising:

logic code for parsing said server response for any broadcast message;

logic code for accessing and updating said gateway database if said server response includes a broadcast message; and

logic code for sending a broadcast response to said server.

36. (Original) The computer program product of claim 19, further comprising:

logic code for receiving a status check request for said set of files;

logic code for downloading a current version of said set of files from a server;

logic code for comparing said current version of said set of files to said set of files;

logic code for generating said status based on said comparing;

logic code for creating a status check response, said status check response including said status of said set of files; and

logic code for sending said status check response.